

1/13

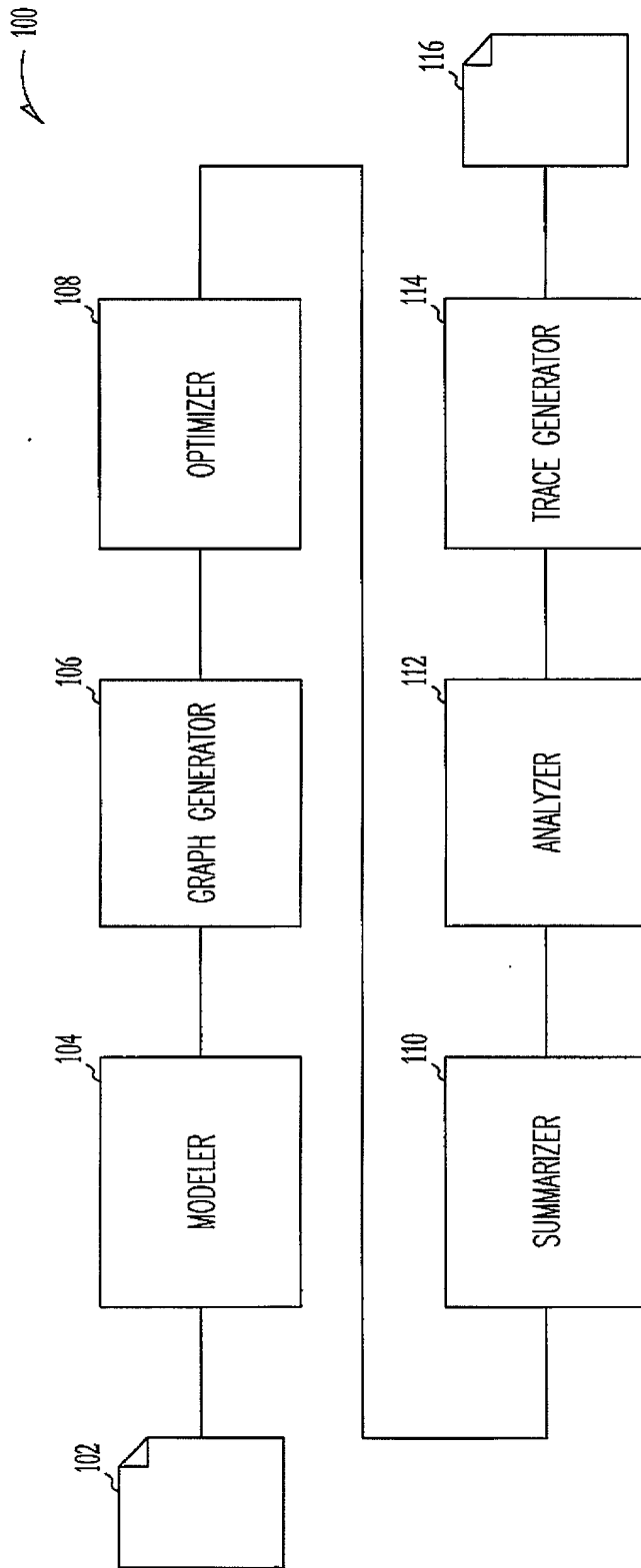


Fig.1

2/13

00 — A

```

int numUnits;
int level;
void getUnit() {
    [1] bool canEnter := F;
    [2] if (numUnits = 0) {
    [3]   if (level > 10) {
    [4]     NewUnit();
    [5]     numUnits := 1;
    [6]     canEnter := T;
    [7]   } else
    [8]     canEnter := T;
    [9]   if (canEnter)
    [10]    if (numUnits = 0) {
    [11]      assert(F);
    [12]    } else
    [13]      gotUnit();
    [14] }
    [15] }

    [16] bool nU0;
    [17] void getUnit() {
    [18]   ...;
    [19]   if (nU0) {
    [20]     if (?) {
    [21]       ...;
    [22]       nU0 := F;
    [23]       cE := T;
    [24]     }
    [25]   } else
    [26]     cE := T;
    [27] }

    [28] bool nU0;
    [29] void getUnit() {
    [30]   ...;
    [31]   if (nU0) {
    [32]     if (?) {
    [33]       ...;
    [34]       nU0 := F;
    [35]       cE := T;
    [36]     }
    [37]   } else
    [38]     ...;
    [39] }

    [40] bool nU0;
    [41] void getUnit() {
    [42]   ...;
    [43]   if (nU0) {
    [44]     if (?) {
    [45]       ...;
    [46]       nU0 := F;
    [47]       cE := T;
    [48]     }
    [49]   } else
    [50]     ...;
    [51] }

    [52] bool nU0;
    [53] void getUnit() {
    [54]   ...;
    [55]   if (nU0) {
    [56]     if (?) {
    [57]       ...;
    [58]       nU0 := F;
    [59]       cE := T;
    [60]     }
    [61]   } else
    [62]     ...;
    [63] }

    [64] bool nU0;
    [65] void getUnit() {
    [66]   ...;
    [67]   if (nU0) {
    [68]     if (?) {
    [69]       ...;
    [70]       nU0 := F;
    [71]       cE := T;
    [72]     }
    [73]   } else
    [74]     ...;
    [75] }

    [76] bool nU0;
    [77] void getUnit() {
    [78]   ...;
    [79]   if (nU0) {
    [80]     if (?) {
    [81]       ...;
    [82]       nU0 := F;
    [83]       cE := T;
    [84]     }
    [85]   } else
    [86]     ...;
    [87] }

    [88] bool nU0;
    [89] void getUnit() {
    [90]   ...;
    [91]   if (nU0) {
    [92]     if (?) {
    [93]       ...;
    [94]       nU0 := F;
    [95]       cE := T;
    [96]     }
    [97]   } else
    [98]     ...;
    [99] }

    [100] bool nU0;
    [101] void getUnit() {
    [102]   ...;
    [103]   if (nU0) {
    [104]     if (?) {
    [105]       ...;
    [106]       nU0 := F;
    [107]       cE := T;
    [108]     }
    [109]   } else
    [110]     ...;
    [111] }

    [112] bool nU0;
    [113] void getUnit() {
    [114]   ...;
    [115]   if (nU0) {
    [116]     if (?) {
    [117]       ...;
    [118]       nU0 := F;
    [119]       cE := T;
    [120]     }
    [121]   } else
    [122]     ...;
    [123] }

    [124] bool nU0;
    [125] void getUnit() {
    [126]   ...;
    [127]   if (nU0) {
    [128]     if (?) {
    [129]       ...;
    [130]       nU0 := F;
    [131]       cE := T;
    [132]     }
    [133]   } else
    [134]     ...;
    [135] }

    [136] bool nU0;
    [137] void getUnit() {
    [138]   ...;
    [139]   if (nU0) {
    [140]     if (?) {
    [141]       ...;
    [142]       nU0 := F;
    [143]       cE := T;
    [144]     }
    [145]   } else
    [146]     ...;
    [147] }

    [148] bool nU0;
    [149] void getUnit() {
    [150]   ...;
    [151]   if (nU0) {
    [152]     if (?) {
    [153]       ...;
    [154]       nU0 := F;
    [155]       cE := T;
    [156]     }
    [157]   } else
    [158]     ...;
    [159] }

    [160] bool nU0;
    [161] void getUnit() {
    [162]   ...;
    [163]   if (nU0) {
    [164]     if (?) {
    [165]       ...;
    [166]       nU0 := F;
    [167]       cE := T;
    [168]     }
    [169]   } else
    [170]     ...;
    [171] }

    [172] bool nU0;
    [173] void getUnit() {
    [174]   ...;
    [175]   if (nU0) {
    [176]     if (?) {
    [177]       ...;
    [178]       nU0 := F;
    [179]       cE := T;
    [180]     }
    [181]   } else
    [182]     ...;
    [183] }

    [184] bool nU0;
    [185] void getUnit() {
    [186]   ...;
    [187]   if (nU0) {
    [188]     if (?) {
    [189]       ...;
    [190]       nU0 := F;
    [191]       cE := T;
    [192]     }
    [193]   } else
    [194]     ...;
    [195] }

    [196] bool nU0;
    [197] void getUnit() {
    [198]   ...;
    [199]   if (nU0) {
    [200]     if (?) {
    [201]       ...;
    [202]       nU0 := F;
    [203]       cE := T;
    [204]     }
    [205]   } else
    [206]     ...;
    [207] }

    [208] bool nU0;
    [209] void getUnit() {
    [210]   ...;
    [211]   if (nU0) {
    [212]     if (?) {
    [213]       ...;
    [214]       nU0 := F;
    [215]       cE := T;
    [216]     }
    [217]   } else
    [218]     ...;
    [219] }

    [220] bool nU0;
    [221] void getUnit() {
    [222]   ...;
    [223]   if (nU0) {
    [224]     if (?) {
    [225]       ...;
    [226]       nU0 := F;
    [227]       cE := T;
    [228]     }
    [229]   } else
    [230]     ...;
    [231] }

    [232] bool nU0;
    [233] void getUnit() {
    [234]   ...;
    [235]   if (nU0) {
    [236]     if (?) {
    [237]       ...;
    [238]       nU0 := F;
    [239]       cE := T;
    [240]     }
    [241]   } else
    [242]     ...;
    [243] }

    [244] bool nU0;
    [245] void getUnit() {
    [246]   ...;
    [247]   if (nU0) {
    [248]     if (?) {
    [249]       ...;
    [250]       nU0 := F;
    [251]       cE := T;
    [252]     }
    [253]   } else
    [254]     ...;
    [255] }

    [256] bool nU0;
    [257] void getUnit() {
    [258]   ...;
    [259]   if (nU0) {
    [260]     if (?) {
    [261]       ...;
    [262]       nU0 := F;
    [263]       cE := T;
    [264]     }
    [265]   } else
    [266]     ...;
    [267] }

    [268] bool nU0;
    [269] void getUnit() {
    [270]   ...;
    [271]   if (nU0) {
    [272]     if (?) {
    [273]       ...;
    [274]       nU0 := F;
    [275]       cE := T;
    [276]     }
    [277]   } else
    [278]     ...;
    [279] }

    [280] bool nU0;
    [281] void getUnit() {
    [282]   ...;
    [283]   if (nU0) {
    [284]     if (?) {
    [285]       ...;
    [286]       nU0 := F;
    [287]       cE := T;
    [288]     }
    [289]   } else
    [290]     ...;
    [291] }

    [292] bool nU0;
    [293] void getUnit() {
    [294]   ...;
    [295]   if (nU0) {
    [296]     if (?) {
    [297]       ...;
    [298]       nU0 := F;
    [299]       cE := T;
    [300]     }
    [301]   } else
    [302]     ...;
    [303] }

    [304] bool nU0;
    [305] void getUnit() {
    [306]   ...;
    [307]   if (nU0) {
    [308]     if (?) {
    [309]       ...;
    [310]       nU0 := F;
    [311]       cE := T;
    [312]     }
    [313]   } else
    [314]     ...;
    [315] }

    [316] bool nU0;
    [317] void getUnit() {
    [318]   ...;
    [319]   if (nU0) {
    [320]     if (?) {
    [321]       ...;
    [322]       nU0 := F;
    [323]       cE := T;
    [324]     }
    [325]   } else
    [326]     ...;
    [327] }

    [328] bool nU0;
    [329] void getUnit() {
    [330]   ...;
    [331]   if (nU0) {
    [332]     if (?) {
    [333]       ...;
    [334]       nU0 := F;
    [335]       cE := T;
    [336]     }
    [337]   } else
    [338]     ...;
    [339] }

    [340] bool nU0;
    [341] void getUnit() {
    [342]   ...;
    [343]   if (nU0) {
    [344]     if (?) {
    [345]       ...;
    [346]       nU0 := F;
    [347]       cE := T;
    [348]     }
    [349]   } else
    [350]     ...;
    [351] }

    [352] bool nU0;
    [353] void getUnit() {
    [354]   ...;
    [355]   if (nU0) {
    [356]     if (?) {
    [357]       ...;
    [358]       nU0 := F;
    [359]       cE := T;
    [360]     }
    [361]   } else
    [362]     ...;
    [363] }

    [364] bool nU0;
    [365] void getUnit() {
    [366]   ...;
    [367]   if (nU0) {
    [368]     if (?) {
    [369]       ...;
    [370]       nU0 := F;
    [371]       cE := T;
    [372]     }
    [373]   } else
    [374]     ...;
    [375] }

    [376] bool nU0;
    [377] void getUnit() {
    [378]   ...;
    [379]   if (nU0) {
    [380]     if (?) {
    [381]       ...;
    [382]       nU0 := F;
    [383]       cE := T;
    [384]     }
    [385]   } else
    [386]     ...;
    [387] }

    [388] bool nU0;
    [389] void getUnit() {
    [390]   ...;
    [391]   if (nU0) {
    [392]     if (?) {
    [393]       ...;
    [394]       nU0 := F;
    [395]       cE := T;
    [396]     }
    [397]   } else
    [398]     ...;
    [399] }

    [400] bool nU0;
    [401] void getUnit() {
    [402]   ...;
    [403]   if (nU0) {
    [404]     if (?) {
    [405]       ...;
    [406]       nU0 := F;
    [407]       cE := T;
    [408]     }
    [409]   } else
    [410]     ...;
    [411] }

    [412] bool nU0;
    [413] void getUnit() {
    [414]   ...;
    [415]   if (nU0) {
    [416]     if (?) {
    [417]       ...;
    [418]       nU0 := F;
    [419]       cE := T;
    [420]     }
    [421]   } else
    [422]     ...;
    [423] }

    [424] bool nU0;
    [425] void getUnit() {
    [426]   ...;
    [427]   if (nU0) {
    [428]     if (?) {
    [429]       ...;
    [430]       nU0 := F;
    [431]       cE := T;
    [432]     }
    [433]   } else
    [434]     ...;
    [435] }

    [436] bool nU0;
    [437] void getUnit() {
    [438]   ...;
    [439]   if (nU0) {
    [440]     if (?) {
    [441]       ...;
    [442]       nU0 := F;
    [443]       cE := T;
    [444]     }
    [445]   } else
    [446]     ...;
    [447] }

    [448] bool nU0;
    [449] void getUnit() {
    [450]   ...;
    [451]   if (nU0) {
    [452]     if (?) {
    [453]       ...;
    [454]       nU0 := F;
    [455]       cE := T;
    [456]     }
    [457]   } else
    [458]     ...;
    [459] }

    [460] bool nU0;
    [461] void getUnit() {
    [462]   ...;
    [463]   if (nU0) {
    [464]     if (?) {
    [465]       ...;
    [466]       nU0 := F;
    [467]       cE := T;
    [468]     }
    [469]   } else
    [470]     ...;
    [471] }

    [472] bool nU0;
    [473] void getUnit() {
    [474]   ...;
    [475]   if (nU0) {
    [476]     if (?) {
    [477]       ...;
    [478]       nU0 := F;
    [479]       cE := T;
    [480]     }
    [481]   } else
    [482]     ...;
    [483] }

    [484] bool nU0;
    [485] void getUnit() {
    [486]   ...;
    [487]   if (nU0) {
    [488]     if (?) {
    [489]       ...;
    [490]       nU0 := F;
    [491]       cE := T;
    [492]     }
    [493]   } else
    [494]     ...;
    [495] }

    [496] bool nU0;
    [497] void getUnit() {
    [498]   ...;
    [499]   if (nU0) {
    [500]     if (?) {
    [501]       ...;
    [502]       nU0 := F;
    [503]       cE := T;
    [504]     }
    [505]   } else
    [506]     ...;
    [507] }

    [508] bool nU0;
    [509] void getUnit() {
    [510]   ...;
    [511]   if (nU0) {
    [512]     if (?) {
    [513]       ...;
    [514]       nU0 := F;
    [515]       cE := T;
    [516]     }
    [517]   } else
    [518]     ...;
    [519] }

    [520] bool nU0;
    [521] void getUnit() {
    [522]   ...;
    [523]   if (nU0) {
    [524]     if (?) {
    [525]       ...;
    [526]       nU0 := F;
    [527]       cE := T;
    [528]     }
    [529]   } else
    [530]     ...;
    [531] }

    [532] bool nU0;
    [533] void getUnit() {
    [534]   ...;
    [535]   if (nU0) {
    [536]     if (?) {
    [537]       ...;
    [538]       nU0 := F;
    [539]       cE := T;
    [540]     }
    [541]   } else
    [542]     ...;
    [543] }

    [544] bool nU0;
    [545] void getUnit() {
    [546]   ...;
    [547]   if (nU0) {
    [548]     if (?) {
    [549]       ...;
    [550]       nU0 := F;
    [551]       cE := T;
    [552]     }
    [553]   } else
    [554]     ...;
    [555] }

    [556] bool nU0;
    [557] void getUnit() {
    [558]   ...;
    [559]   if (nU0) {
    [560]     if (?) {
    [561]       ...;
    [562]       nU0 := F;
    [563]       cE := T;
    [564]     }
    [565]   } else
    [566]     ...;
    [567] }

    [568] bool nU0;
    [569] void getUnit() {
    [570]   ...;
    [571]   if (nU0) {
    [572]     if (?) {
    [573]       ...;
    [574]       nU0 := F;
    [575]       cE := T;
    [576]     }
    [577]   } else
    [578]     ...;
    [579] }

    [580] bool nU0;
    [581] void getUnit() {
    [582]   ...;
    [583]   if (nU0) {
    [584]     if (?) {
    [585]       ...;
    [586]       nU0 := F;
    [587]       cE := T;
    [588]     }
    [589]   } else
    [590]     ...;
    [591] }

    [592] bool nU0;
    [593] void getUnit() {
    [594]   ...;
    [595]   if (nU0) {
    [596]     if (?) {
    [597]       ...;
    [598]       nU0 := F;
    [599]       cE := T;
    [600]     }
    [601]   } else
    [602]     ...;
    [603] }

    [604] bool nU0;
    [605] void getUnit() {
    [606]   ...;
    [607]   if (nU0) {
    [608]     if (?) {
    [609]       ...;
    [610]       nU0 := F;
    [611]       cE := T;
    [612]     }
    [613]   } else
    [614]     ...;
    [615] }

    [616] bool nU0;
    [617] void getUnit() {
    [618]   ...;
    [619]   if (nU0) {
    [620]     if (?) {
    [621]       ...;
    [622]       nU0 := F;
    [623]       cE := T;
    [624]     }
    [625]   } else
    [626]     ...;
    [627] }

    [628] bool nU0;
    [629] void getUnit() {
    [630]   ...;
    [631]   if (nU0) {
    [632]     if (?) {
    [633]       ...;
    [634]       nU0 := F;
    [635]       cE := T;
    [636]     }
    [637]   } else
    [638]     ...;
    [639] }

    [640] bool nU0;
    [641] void getUnit() {
    [642]   ...;
    [643]   if (nU0) {
    [644]     if (?) {
    [645]       ...;
    [646]       nU0 := F;
    [647]       cE := T;
    [648]     }
    [649]   } else
    [650]     ...;
    [651] }

    [652] bool nU0;
    [653] void getUnit() {
    [654]   ...;
    [655]   if (nU0) {
    [656]     if (?) {
    [657]       ...;
    [658]       nU0 := F;
    [659]       cE := T;
    [660]     }
    [661]   } else
    [662]     ...;
    [663] }

    [664] bool nU0;
    [665] void getUnit() {
    [666]   ...;
    [667]   if (nU0) {
    [668]     if (?) {
    [669]       ...;
    [670]       nU0 := F;
    [671]       cE := T;
    [672]     }
    [673]   } else
    [674]     ...;
    [675] }

    [676] bool nU0;
    [677] void getUnit() {
    [678]   ...;
    [679]   if (nU0) {
    [680]     if (?) {
    [681]       ...;
    [682]       nU0 := F;
    [683]       cE := T;
    [684]     }
    [685]   } else
    [686]     ...;
    [687] }

    [688] bool nU0;
    [689] void getUnit() {
    [690]   ...;
    [691]   if (nU0) {
    [692]     if (?) {
    [693]       ...;
    [694]       nU0 := F;
    [695]       cE := T;
    [696]     }
    [697]   } else
    [698]     ...;
    [699] }

    [700] bool nU0;
    [701] void getUnit() {
    [702]   ...;
    [703]   if (nU0) {
    [704]     if (?) {
    [705]       ...;
    [706]       nU0 := F;
    [707]       cE := T;
    [708]     }
    [709]   } else
    [710]     ...;
    [711] }

    [712] bool nU0;
    [713] void getUnit() {
    [714]   ...;
    [715]   if (nU0) {
    [716]     if (?) {
    [717]       ...;
    [718]       nU0 := F;
    [719]       cE := T;
    [720]     }
    [721]   } else
    [722]     ...;
    [723] }

    [724] bool nU0;
    [725] void getUnit() {
    [726]   ...;
    [727]   if (nU0) {
    [728]     if (?) {
    [729]       ...;
    [730]       nU0 := F;
    [731]       cE := T;
    [732]     }
    [733]   } else
    [734]     ...;
    [735] }

    [736] bool nU0;
    [737] void getUnit() {
    [738]   ...;
    [739]   if (nU0) {
    [740]     if (?) {
    [741]       ...;
    [742]       nU0 := F;
    [743]       cE := T;
    [744]     }
    [745]   } else
    [746]     ...;
    [747] }

    [748] bool nU0;
    [749] void getUnit() {
    [750]   ...;
    [751]   if (nU0) {
    [752]     if (?) {
    [753]       ...;
    [754]       nU0 := F;
    [755]       cE := T;
    [756]     }
    [757]   } else
    [758]     ...;
    [759] }

    [760] bool nU0;
    [761] void getUnit() {
    [762]   ...;
    [763]   if (nU0) {
    [764]     if (?) {
    [765]       ...;
    [766]       nU0 := F;
    [767]       cE := T;
    [768]     }
    [769]   } else
    [770]     ...;
    [771] }

    [772] bool nU0;
    [773] void getUnit() {
    [774]   ...;
    [775]   if (nU0) {
    [776]     if (?) {
    [777]       ...;
    [778]       nU0 := F;
    [779]       cE := T;
    [780]     }
    [781]   } else
    [782]     ...;
    [783] }

    [784] bool nU0;
    [785] void getUnit() {
    [786]   ...;
    [787]   if (nU0) {
    [788]     if (?) {
    [789]       ...;
    [790]       nU0 := F;
    [791]       cE := T;
    [792]     }
    [793]   } else
    [794]     ...;
    [795] }

    [796] bool nU0;
    [797] void getUnit() {
    [798]   ...;
    [799]   if (nU0) {
    [800]     if (?) {
    [801]       ...;
    [802]       nU0 := F;
    [803]       cE := T;
    [804]     }
    [805]   } else
    [806]     ...;
    [807] }

    [808] bool nU0;
    [809] void getUnit() {
    [810]   ...;
    [811]   if (nU0) {
    [812]     if (?) {
    [813]       ...;
    [814]       nU0 := F;
    [815]       cE := T;
    [816]     }
    [817]   } else
    [818]     ...;
    [819] }

    [820] bool nU0;
    [821] void getUnit() {
    [822]   ...;
    [823]   if (nU0) {
    [824]     if (?) {
    [825]       ...;
    [826]       nU0 := F;
    [827]       cE := T;
    [828]     }
    [829]   } else
    [830]     ...;
    [831] }

    [832] bool nU0;
    [833] void getUnit() {
    [834]   ...;
    [835]   if (nU0) {
    [836]     if (?) {
    [837]       ...;
    [838]       nU0 := F;
    [839]       cE := T;
    [840]     }
    [841]   } else
    [842]     ...;
    [843] }

    [844] bool nU0;
    [845] void getUnit() {
    [846]   ...;
    [847]   if (nU0) {
    [848]     if (?) {
    [849]       ...;
    [850]       nU0 := F;
    [851]       cE := T;
    [852]     }
    [853]   } else
    [854]     ...;
    [855] }

    [856] bool nU0;
    [857] void getUnit() {
    [858]   ...;
    [859]   if (nU0) {
    [860]     if (?) {
    [861]       ...;
    [862]       nU0 := F;
    [863]       cE := T;
    [864]     }
    [865]   } else
    [866]     ...;
    [867] }

    [868] bool nU0;
    [869] void getUnit() {
    [870]   ...;
    [871]   if (nU0) {
    [872]     if (?) {
    [873]       ...;
    [874]       nU0 := F;
    [875]       cE := T;
    [876]     }
    [877]   } else
    [878]     ...;
    [879] }

    [880] bool nU0;
    [881] void getUnit() {
    [882]   ...;
    [883]   if (nU0) {
    [884]     if (?) {
    [885]       ...;
    [886]       nU0 := F;
    [887]       cE := T;
    [888]     }
    [889]   } else
    [890]     ...;
    [891] }

    [892] bool nU0;
    [893] void getUnit() {
    [894]   ...;
    [895]   if (nU0) {
    [896]     if (?) {
    [897]       ...;
    [898]       nU0 := F;
    [899]       cE := T;
    [900]     }
    [901]   } else
    [902]     ...;
    [903] }

    [904] bool nU0;
    [905] void getUnit() {
    [906]   ...;
    [907]   if (nU0) {
    [908]     if (?) {
    [909]       ...;
    [910]       nU0 := F;
    [911]       cE := T;
    [912]     }
    [913]   } else
    [914]     ...;
    [915] }

    [916] bool nU0;
    [917] void getUnit() {
    [918]   ...;
    [919]   if (nU0) {
    [920]     if (?) {
    [921]       ...;
    [922]       nU0 := F;
    [923]       cE := T;
    [924]     }
    [925]   } else
    [926]     ...;
    [927] }

    [928] bool nU0;
    [929] void getUnit() {
    [930]   ...;
    [931]   if (nU0) {
    [932]     if (?) {
    [933]       ...;
    [934]       nU0 := F;
    [935]       cE := T;
    [936]     }
    [937]   } else
    [938]     ...;
    [939] }

    [940] bool nU0;
    [941] void getUnit() {
    [942]   ...;
    [943]   if (nU0) {
    [944]     if (?) {
    [945]       ...;
    [946]       nU0 := F;
    [947]       cE := T;
    [948]     }
    [949]   } else
    [950]     ...;
    [951] }

    [952] bool nU0;
    [953] void getUnit() {
    [954]   ...;
    [955]   if (nU0) {
    [956]     if (?) {
    [957]       ...;
    [958]       nU0 := F;
    [959]       cE := T;
    [960]     }
    [961]   } else
    [962]     ...;
    [963] }

    [964] bool nU0;
    [965] void getUnit() {
    [966]   ...;
    [967]   if (nU0) {
    [968]     if (?) {
    [969]       ...;
    [970]       nU0 := F;
    [971]       cE := T;
    [972]     }
    [973]   } else
    [974]     ...;
    [975] }

    [976] bool nU0;
    [977] void getUnit() {
    [978]   ...;
    [979]   if (nU0) {
    [980]     if (?) {
    [981]       ...;
    [982]       nU0 := F;
    [983]       cE := T;
    [984]     }
    [985]   } else
    [986]     ...;
    [987] }

    [988] bool nU0;
    [989] void getUnit() {
    [990]   ...;
    [991]   if (nU0) {
    [992]     if (?) {
    [993]       ...;
    [994]       nU0 := F;
    [995]       cE := T;
    [996]     }
    [997]   } else
    [998]     ...;
    [999] }

    [1000] bool nU0;
    [1001] void getUnit() {
    [1002]   ...;
    [1003]   if (nU0) {
    [1004]     if (?) {
    [1005]       ...;
    [1006]       nU0 := F;
    [1007]       cE := T;
    [1008]     }
    [1009]   } else
    [1010]     ...;
    [1011] }

    [1012] bool nU0;
    [1013] void getUnit() {
    [1014]   ...;
    [1015]   if (nU0) {
    [1016]     if (?) {
    [1017]       ...;
    [1018]       nU0 := F;
    [1019]       cE := T;
    [1020]     }
    [1021]   } else
    [1022]     ...;
    [1023] }

    [1024] bool nU0;
    [1025] void getUnit() {
    [1026]   ...;
    [1027]   if (nU0) {
    [1028]     if (?) {
    [1029]       ...;
    [1030]       nU0 := F;
    [1031]       cE := T;
    [1032]     }
    [1033]   } else
    [1034]     ...;
    [1035] }

    [1036] bool nU0;
    [1037] void getUnit() {
    [1038]   ...;
    [1039]   if (nU0) {
    [1040]     if (?) {
    [1041]       ...;
    [1042]       nU0 := F;
    [1043]       cE := T;
    [1044]     }
    [1045]   } else
    [1046]     ...;
    [1047] }

    [1048] bool nU0;
    [1049] void getUnit() {
    [1050]   ...;
    [1051]   if (nU0) {
    [1052]     if (?) {
    [1053]       ...;
    [1054]       nU0 := F;
    [1055]       cE := T;
    [1056]     }
    [1057]   } else
    [1058]     ...;
    [1059] }

    [1060] bool nU0;
    [1061] void getUnit() {
    [1062]   ...;
    [1063]   if (nU0) {
    [1064]     if (?) {
    [1065]       ...;
    [1066]       nU0 := F;
    [1067]       cE := T;
    [1068]     }
    [1069]   } else
    [1070]     ...;
    [1071] }

    [1072] bool nU0;
    [1073] void getUnit() {
    [1074]   ...;
    [1075]   if (nU0) {
    [1076]     if (?) {
    [1077]       ...;
    [1078]       nU0 := F;
    [1079]       cE := T;
    [1080]     }
    [1081]   } else
    [1082]     ...;
    [1083] }

    [1084] bool nU0;
    [1085] void getUnit() {
    [1086]   ...;
    [1087
```

302

304

```

Line 12          State g=1 h=0
Line 11          State g=1 h=0
Line 10          State g=1 h=0
      Line 22     State g=1 a1=1 a2=0
      Line 24     State g=1 a1=0 a2=1
      Line 20     State g=1 a1=0 a2=1
      Line 21     State g=1 a1=1 a2=0
      Line 20     State g=1 a1=1 a2=0
Line 9           State g=1 h=0
Line 8           State g=1 h=0
      Line 22     State g=1 a1=1 a2=0
      Line 24     State g=1 a1=0 a2=1
      Line 20     State g=1 a1=0 a2=1
      Line 21     State g=1 a1=1 a2=0
      Line 20     State g=1 a1=1 a2=0
Line 7           State g=1 h=0
Line 6           State g=1

```

Fig.3

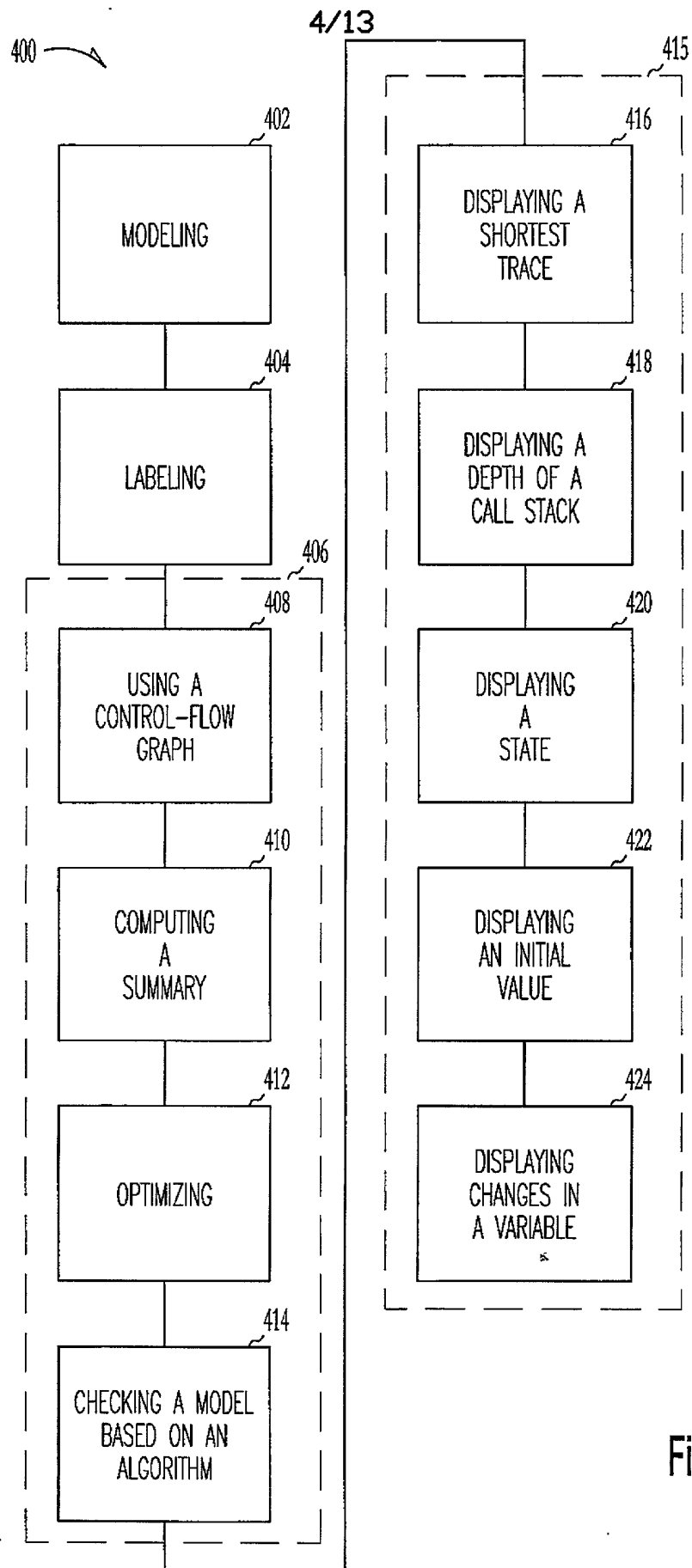


Fig.4

5/13

500

```
[1]  if (z) {
[2]    x:=1;
      else
[3]    x:=z;
[4]    z:=y|x;
```

Fig.5A

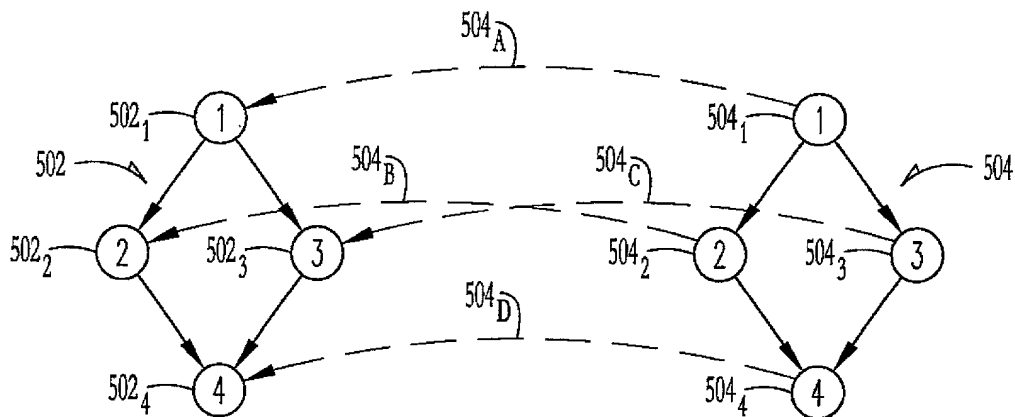


Fig.5B

6/13

600

```
Boolean g;  
main() {  
  [2]   if (z) {  
  [3]     x:=1;  
  [4]     else  
          x:=0;  
  [5]     z:=y+x;  
  [6]     foo (z);  
  [7]     skip;  
  [8]   }  
  [9]   foo (z) {  
  [10]    g:=1;  
  [11]  }  
}
```

Fig.6A

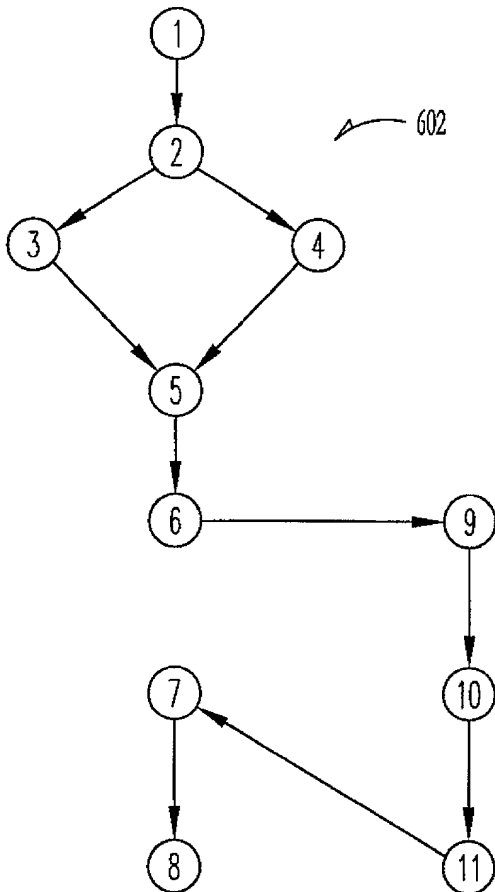


Fig.6B

7/13

700

v	$Transfer_v$
702 skip print goto return	$\lambda(\Omega_1, \Omega_2).(\Omega_2 = \Omega_1)$
704 $x_1, \dots, x_k :=$ e_1, \dots, e_k	$\lambda(\Omega_1, \Omega_2).(\Omega_2 = \Omega_1 [x_1/\Omega_1(e_1)] \dots [x_k/\Omega_1(e_k)])$
706 if(d) while(d) assert(d)	$Transfer_{v,true} = \lambda(\Omega_1, \Omega_2).((\Omega_1(d) = 1) \wedge (\Omega_2 = \Omega_1))$ $Transfer_{v,false} = \lambda(\Omega_1, \Omega_2).((\Omega_1(d) = 0) \wedge (\Omega_2 = \Omega_1))$
708 pr(e_1, \dots, e_k)	$\lambda(\Omega_1, \Omega_2).(\Omega_2 = \Omega_1 [x_1/\Omega_1(e_1)] \dots [x_k/\Omega_1(e_k)]),$ where x_1, \dots, x_k are the formal parameters of pr

Fig.7

0986090501 06098860

8/13

800

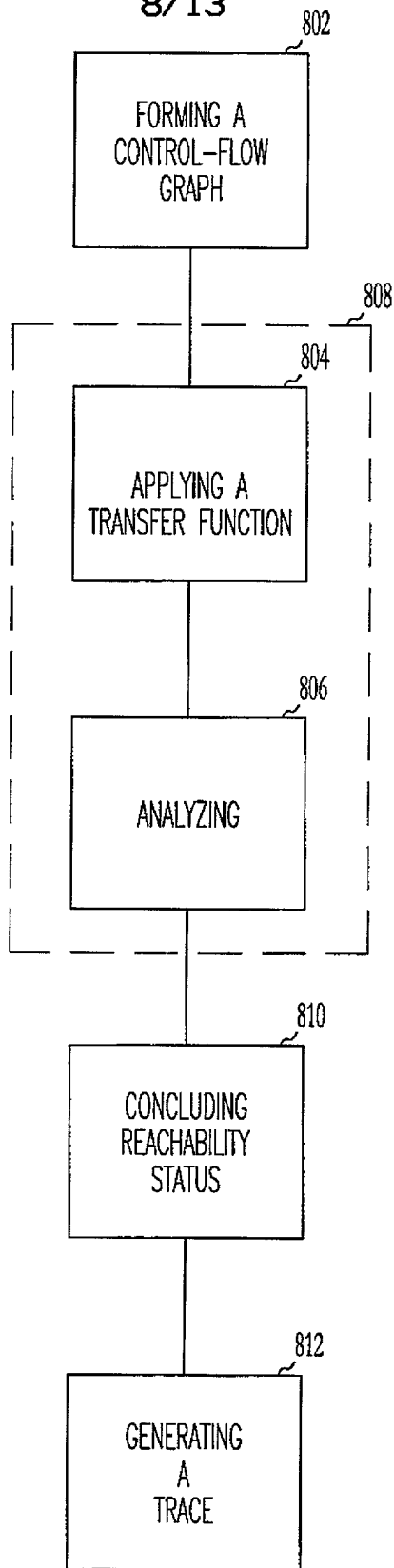


Fig.8

9/13

```

902 global PathEdges, SummaryEdges, WorkList 900
904 procedure Propagate(v, p)
    begin
906 if  $p \not\subseteq \text{PathEdges}(v)$  then
908  $\text{PathEdges}(v) := \text{PathEdges}(v) \cup p$ 
910 Insert  $v$  into WorkList fi
    fi
912 end

914 procedure Reachable( $G_B$ )
    begin
916 for all  $v \in V_B$  do  $\text{PathEdges}(v) := \{\}$ 
917 for all  $v \in \text{Call}_B$  do  $\text{SummaryEdges}(v) := \{\}$ 
918  $\text{PathEdges}(\text{First}_B(\text{main})) :=$ 
         $\{\langle \Omega, \Omega \rangle \mid \Omega \text{ is any valuation to globals and local variables of main} \}$ 
920  $\text{WorkList} := \{\text{First}_B(\text{main})\}$ 
922 while  $\text{WorkList} \neq 0$  do
924 remove vertex  $v$  from WorkList
926 switch ( $v$ )
928 case  $v \in \text{Call}_B$ 
         $\text{Propagate}(\text{sSucc}_B(v), \text{SelfLoop}(\text{Join}(\text{PathEdges}(v), \text{Transfer}_v)))$  930
         $\text{Propagate}(\text{ReturnPt}_B(v), \text{Join}(\text{PathEdges}(v), \text{SummaryEdges}(v)))$  932
934 case  $v \in \text{Exit}_B$ :
        for each  $w \in \text{Succ}_B(v)$  do 936
            let
                 $c \in \text{Call}_B$  such that  $w = \text{ReturnPt}_B(c)$  and 938
                 $s = \text{Lift}_c(\text{PathEdges}(v), \text{ProcOf}_B(v))$  940
            in
                if  $s \not\subseteq \text{SummaryEdges}(c)$  then 944
                     $\text{SummaryEdges}(c) := \text{SummaryEdges}(c) \cup s$  946
                     $\text{Propagate}(w, \text{Join}(\text{PathEdges}(c), \text{SummaryEdges}(c)))$ ; 948
                ni
950 case  $v \in \text{Cond}_B$ :
                 $\text{Propagate}(\text{Tsucc}_B(v), \text{Join}(\text{PathEdges}(v), \text{Transfer}_{v, \text{true}}))$  952
                 $\text{Propagate}(\text{Fsucc}_B(v), \text{Join}(\text{PathEdges}(v), \text{Transfer}_{v, \text{false}}))$  954
956 case  $v \in V_B - \text{Call}_B - \text{Exit}_B - \text{Cond}_B$ :
                let  $p = \text{Join}(\text{PathEdges}(v), \text{Transfer}_v)$  in 958
                for each  $w \in \text{Succ}_B(v)$  do 960
                     $\text{Propagate}(w, p)$  962
                ni
    end

```

Fig.9

FOUO 06099860

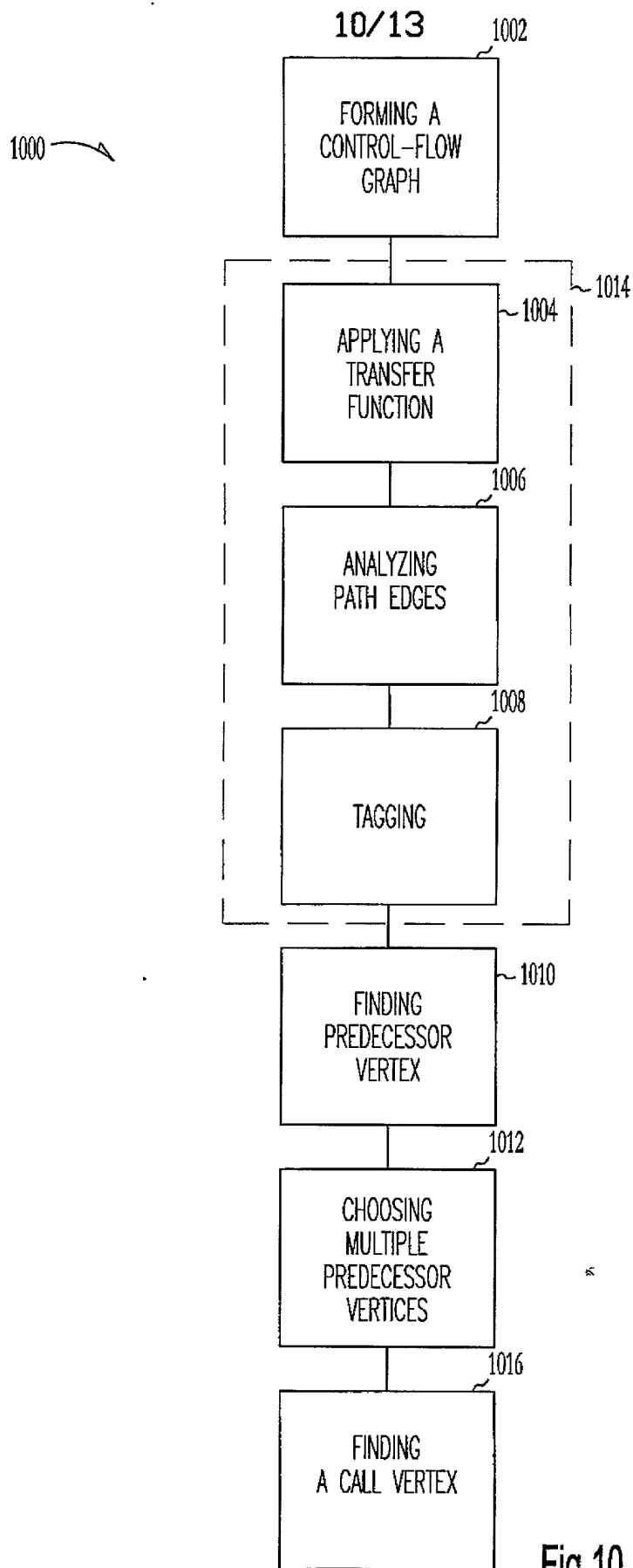


Fig.10

11/13

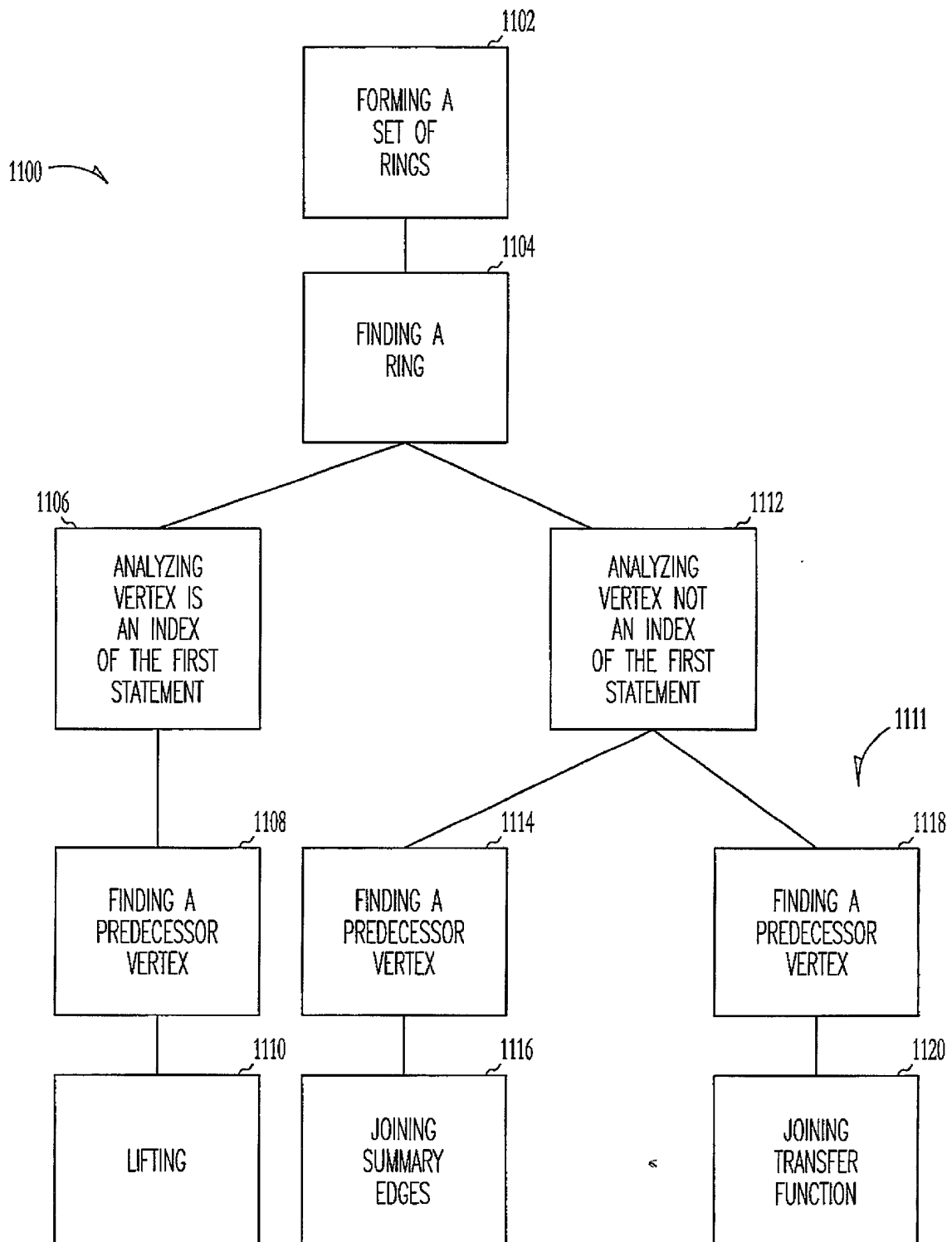


Fig.11

12/13

global

$PE' : V_G \rightarrow \text{set-of } (D \times D)$

$Worklist : V_G \rightarrow \text{set-of } (D \times D)$

procedure Propagate($v : V_{G,p} : (D \times D)$)

begin

if $p \notin PE'(v)$ **then**

$PE'(v) := PE'(v) \cup \{p\}$

$Worklist(v) := Worklist(v) \cup \{p\}$

fi

end

procedure CMOP_{SP_{rhs}}($S : \text{set-of } D$)

begin

$PE'(\text{entry}) := \{\langle d, d \rangle \mid d \in S\}$

$Worklist(\text{entry}) := PE'(\text{entry})$

while $\exists v_2 \text{ s.t. } Worklist(v_2) \neq \emptyset$ **do**

 select and remove $\langle d_1, d_2 \rangle$ from $Worklist(v_2)$

for each $v_2 \rightarrow v_3 \in E_G$ **do**

for each $d_3 \in M(v_2 \rightarrow v_3)(\{d_2\})$ **do**

 Propagate($v_3 \langle d_1, d_3 \rangle$)

od

od

od

end

Fig.12

13/13

global
 $PE' : V \rightarrow \text{set-of}(\text{set-of } D \times \text{set-of } D)$
 $Worklist : V_G \rightarrow \text{set-of}(\text{set-of } D \times \text{set-of } D)$

procedure Propagate($v; V_G, p : (\text{set-of } D \times \text{set-of } D)$)
begin
 if $p \notin PE'(v)$ then
 $PE'(v) := PE'(v) \cup \{p\}$
 $Worklist(v) := Worklist(v) \cup \{p\}$
 fi
end

procedure CSMOP_{SP_{rhs}}($S' : \text{set-of}(\text{set-of } D)$)
begin
 $PE'(\text{entry}) := \{\langle S, S \rangle \mid S \in S'\}$
 $Worklist(\text{entry}) := PE'(\text{entry})$
while $\exists v_2 \text{ s.t. } Worklist(v_2) \neq \emptyset$ **do**
 select and remove $\langle S_1, S_2 \rangle$ from $Worklist(v_2)$
for each $v_2 \rightarrow v_3 \in E_G$ **do**
 let $S_3 = M(v_2 \rightarrow v_3)(S_2)$ **in**
 Propagate($v_3, \langle S_1, S_3 \rangle$)
ni
od
od
end

Fig.13